

2017

Basketmunio

*Autor: Adrián Mateo y Alejandro Villalba*

Desarrollo de Aplicaciones Web 20-06-2017

Basketmunio

Índice

1. **Descripción general del proyecto3**
   1. Objetivos**4**
   2. Motivación del proyecto**5**
   3. Necesidades a cubrir**5**
2. **Metodología6**
3. **Proyecto8**
   1. Estructura del Sistema y componentes**8**
   2. Requisitos del sistema**10**
   3. Diseño**11**
   4. Pruebas a nivel modular y global**14**
   5. Mantenimiento**14**
4. **Conclusiones15**
   1. Objetivos alcanzados**15**
   2. Futuras mejoras**16**
5. **Referencias bibliográficas16**
6. **Anexos18**

1. Descripción general del proyecto

Basketmunio es un Proyecto el cual consiste en un videojuego on-line manager de baloncesto, concretamente de la NBA. Se juega con amigos u otros usuarios aleatorios y requiere un buen conocimiento de la situación actual de cada equipo, ya que se puede hacer una nueva alineación cada semana. También es necesaria la habilidad de administrar la comprar-venta de jugadores. La puntuación del equipo de cada usuario será otorgada en un principio mensualmente y dependerá de la actuación de los jugadores de su plantilla en los partidos reales y oficiales de la temporada regular de la NBA.

Al crear tu equipo se te otorgarán diez jugadores aleatorios de entre los más de cuatrocientos que poseemos, cada uno con sus estadísticas (nueve estadísticas diferentes por jugador), a continuación, tendrás dos opciones, meter a tu equipo en unas de nuestras ligas publicas ya definidas o bien crear tu propia liga y tener a tu equipo en esta.

Podrás contemplar las estadísticas de tus jugadores y mucho más, por ejemplo, máximos anotadores o con mas tapones. También podras seguir nuestros progresos de desarrollo en nustras redes sociales (facebook y twitter) que las puedes encontrar en el pie de cada página.

Si eres un poco despistado podrás enterarte de la programación de partidos de la NBA en nuestro calendario, puediendo elegir el mes de los partidos que quieres ver, y podras observar la fecha y hora, equipos que se enfrentan y el resultado final si ya se ha disputado.

* 1. Objetivos

Los objetivos principales para este proyecto son:

* Estructura del proyecto en modelo vista controlador.
* Páginas de logueo y registro con validación jQuery sobre una BBDD.
* Envío automático de emails de confirmación de registro y de restablecimiento de contraseña.
* Uso de sesiones para controlar la estancia del usuario en la página.
* Creación de BBDD relacional donde almacenar tanto los datos de registros de usuario como toda la información perteneciente a los jugadores y equipos del juego.
* Creación de páginas dinámicas utilizando llamadas AJAX y “*responsive*” utilizando “medias queries” y operadores lógicos para adaptar dichas páginas hasta un minimo de resolución.
* Extracción de los datos en formato “*csv*” de la página oficial de la NBA mediante “*scraping*”.
* Creación de scripts automáticos con los que generar, insertar y actualizar los datos de nuestra BBDD.
* La posibilidad de que un usuario pueda crear un equipo, el cual se inicializará con diez jugadores aleatorios.
* La posibilidad de que un usuario pueda crear o bien unirse a una liga con uno de sus equipos.
* Tener un mercado donde se muestren todos los jugadores que no se encuentren en ningún equipo y poder realizar traspasos, tanto fichajes como ventas de tus propios jugadores.
* La posibilidad de dar a nuestros usuarios un medio de comunicación directo con nosotros, tanto por nuestras redes sociales (que se encuentran a pie de página) o bien por el correo electrónico que se encuentra en la sección de ayuda del juego.
* El usuario podrá comprobar las estadísticas de las plantillas de sus equipos, también podrá observar los jugadores más destacados de estas.
* Se podrá hacer una búsqueda de una liga o bien observar todas las existentes. En estos resultados también se mostrará el numero de usuario que hay en esa liga.
* Apartado en el que el usuario podrá comprobar la clasificación en la liga del equipo que seleccione, en esta se mostrará una clasificación completa con los puntos actuales de cada equipo de la liga.
* Se podrá examinar las estadísticas completas de cada uno de los jugadores del equipo que seleccione el usuario, o bien hacer un examen muy completo observando las estadísticas de todos y cada de uno de los jugadores (que son más de 400).
* Proporcionar a nuestros usuarios una página de ayuda donde se redacta toda la información necesaria para empezar a jugar, o simplemente conocernos o contactar con nosotros mediante el contacto que proporcionamos.
* Que el usuario pueda cerrar la sesión en cualquier momento, y volver de esta manera a la página de login.
  1. Motivación del proyecto

La motivación principal por la que hemos decidido hacer este proyecto, son los numerosos juegos de manager que podemos encontrar a lo largo de internet. Uno de estos, de características similares y ambientado en otro tipo de deporte, fueron los creadores de esta motivación junto a un amigo que nos inspiró y motivo a hacer este proyecto final.

Otros de los motivos de esta motivación fue la escasez de dichos juegos en el ámbito del baloncesto, concretamente en la NBA, en nuestro respectivo idioma.

* 1. Necesidades a cubrir

Este proyecto está exclusivamente orientado para cubrir las necesidades de ocio y entretenimiento, es decir, crear un juego con el que pasar el rato y compertir con tus amigos.

1. Metodología

En este apartado vamos a definir toda la tecnología, herramientas de desarrollo, recursos… que han sido necesario para llevar acabo este proyecto.

Comenzaremos con los lenguajes en los que ha sido desarroda la página, que son los siguientes:

* PHP 5.6.25
* HTML5
* CSS3
* Con respecto a javascript se ha usado lo orientado a:
  + JQuery 3.1.1
  + Objetos JSON
* SQL

A continuación, se mostrará las herramientas que se han utilizado para hacer el desarrollo:

* JetBrains PhpStorm 2017.1.4
* JetBrains PhpStorm 2016.2.1
* JetBrains WebStorm 2016.2.3
* NetBeans IDE 8.1
* Notepad ++ v6.9.2
* Brackets 1.9
* Toad Data Modeler 5.4
* Extensiones de Google (explicadas al final del apartado):
  + Data Miner (para hacer “*scraping*”)
  + Resize Window
  + Json viewer
* Sublime Text 3
* WampServer 3.0.6:
  + Apache 2.4.23
  + PHP 5.3
  + MySql 5.7.14
* Microsoft Office Professional Plus 2016:
  + Word
  + Excel
* GitHub Desktop 0.6.1 (explicación al final del apartado)
* Google Chrome Versión 58.0.3029.110 (64-bit)

En cuanto a las especificaciones hardware que han sido necesarias para la elaboración del proyecto son las siguientes.

|  |  |
| --- | --- |
| Equipo | Basado en 64x ACP |
| SO | Windows 10 Pro |
| Tipo de CPU | QuadCore Intel Core i7-4770K, 3900 MHz |
| Nombre de la placa base | Gigabyte G1.Sniper Z97 |
| Chipset de la placa base | Intel Wildcat Point Z97, Intel Haswell |
| Memoria del sistema (RAM) | 16 GB (DDR3-1333 DDR3 SDRAM) |
| Adaptador video | NVIDIA GeForce GTX 970 (4 GB) |
| Tamaño almacenamiento | 5.8 TB (HDD) - 200GB SSD |

Explicación de las extensiones de Google utilizadas:

* **Data Miner:** extensión utlizada para hacer “*scraping”*, que consiste en extraer información de sitios web, y esta extensión la almacena en ficheros con formato csv.
* **Resize Window:** la funcionalidad de esta es simular la ventana de nuestro navegador en una resolución distinta mas pequeña para poder diseñar la web.
* **Json viewer:** muestra por navegador documentos o respuestas en formato json de forma ordena y clara de entender para el usuario.

**GitHub**

En este proyecto hemos adquirido un repositorio privado de GitHub. GIT es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.

\*Se podrá dar acceso al repositorio (contactarnos).

1. Proyecto
   1. Estructuras del Sistema, diagramas y componentes.

A continuación, pasamos a explicar la estructura en la que esta dividida y se basa el proyecto:

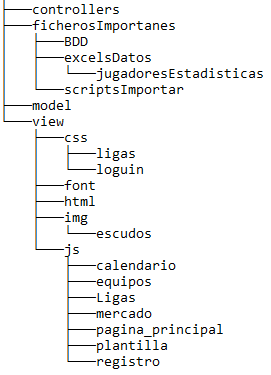
* El modelo que se ha intentado seguir es el MVC (Modelo Vista Controlador), aunque hemos utilizado también técnicas adquiridas en la empresa para la gestión y tratamiento del código.

Árbol de la estructura (solo directorios)

* *basketmunio*
  + *controllers*: en este directorio almacenamos todos los controladores necesarios para el correcto funcionamiento de la página. Son los intermediarios entre el “*model*” y el “*view”.*
  + *ficherosImportantes*: el siguiente directorio lo hemos utilizado para almacenar todos los CSV, scripts de importación de datos y creación para nuestra BBDD.
  + *model*: es el core (núcleo) del proyecto, archivos encargados de realizar consultas a la BBDD y obtener resultados los cuales son devueltos a los controladores.
  + *view*: en esta carpeta se encuentran los ficheros encargados de toda la parte front del proyecto. Este directorio se divide en cinco directorios:
    - *css*: aquí se encuentran todas las hojas de estilos, algunas se encuentran en subdirectorios.
    - *font*: la funcionalidad de esta carpeta es la de mostrar bien los “*favicons*” e iconos de la web.
    - *html:* esta es la carpeta contendora de todas las páginas html que se van a mostrar en la web, en nuestro caso las hemos hecho con formato “*php*” para el correcto funcionamiento de sus funcionalidades.
    - *img*: en esta carpeta se almacenan todas las imágenes usadas en el proyecto.
    - *js*: este es el directorio encargado de almacenar todos los scripts jQuery que son utilizados por las páginas, este también se encuentra ordenado en subdirectorios.
* *index.php*: fichero que se encuentra en cada uno de los directorios del proyecto y que se encarga del redireccionamiento en caso de querer acceder e intentar ver los ficheros que contiene dicho directorio.

Diagrama de la estructura (solo directorios)

Este es el diagrama de la estructura anteriormente explicada que parten del directorio padre llamado “basketmunio”:



Componentes

Esta página no contiene muchos componentes, especialmente contiene dos:

* **Últimos tweets:** es un plugin que concede twitter y que puedes implementar en tus páginas y cuya función es mostrar en esta tus últimos “*tweets*” escritos.
* **Google fonts:** son enlaces que nos ofrece Google para poder importar y utilizar distintos tipos de fuente en tus páginas.
  1. Requisitos del sistema

En este apartado vamos a explicar cuales son los requisitos necesarios tanto para poder jugar como para implementar el proyecto en nuevo “*host”*.

Desde el punto de vista del usuario los requisitos necesarios para el correcto funcionamiento del juego serán muy bajos, ya que solo será necesario un dispositivo con navegador de internet que soporte los distintos lenguajes utilizados, que suelen ser la gran mayoría, y conexión a internet.

En cambio, para poder alojar nuestra web en un host será necesario poseer un equipo (“*host*”) con un mínimo 10GB de almacenamiento y una memoria RAM de al menos 4GB para un correcto y óptimo funcionamiento. Para poder poner en marcha dicha web necesitaremos que el host nombrado anteriormente posea un servidor “*Apache*” o de características similares donde servirá nuestra web. También deberá poseer un interprete de PHP y un sistema de gestión de bases de datos relacionales MySQL.

* 1. Diseño

**Diseño BBDD**

Nuestra base de datos esta compuesta por las siguientes tablas:

-Usuario: Esta es una de las tablas más importantes de nuestra BBDD y

con ella conseguimos almacenar a los usuarios que se registran en nuestra

web en esta encontramos el id\_usuario que se lo asignamos automáticamente con un

autoincrement, apodo\_usuario, nombre\_usuario, apellidos\_usuario, email\_usuario, password\_usuario,fecha\_nacimiento\_usuario toda esta información la introduce el usuario cuando se va a registrar mientras que el codigo\_activación\_usuario lo añadimos mediante un script cuando enviamos el email, activado\_usuario este campo por defecto es 0 si el usuario introduce correctamente el código de activación este campo lo ponemos a 1 y significa que tiene la cuenta activada y por tanto puede acceder a la web.

-Equipos: En esta tabla se encuentran todos los equipos que se crean los usuarios,

el id\_equipo se le asigna automáticamente mediante un script y el id\_usuario lo obtiene como (PFK) mientras que la liga lo obtiene como (FK) Foreing key, nombre y escudo los introduce el usuario mientras que presupuesto le damos un value default de 100 millones y los puntos los mete un script.

-Ligas: En esa tabla se encuentran todas las ligas que se crean por los usuarios y las oficiales, en esta podemos encontrar el id\_liga que se asigna automáticamente mediante un autoincrement, el nombre\_liga, tipo\_liga y contraseña\_liga que los introduce el usuario.

-Jugadores: En esta tabla se encuentran todos los jugadores de la NBA (los 464), es la plantilla base de os jugadores que utiliza cada liga. En ella almacenamos el id\_jugador que se lo asignamos automaticamente con un autoincrement, nombre\_jugador, posicion\_jugador, equipo\_real\_jugador toda esta información la obtenemosde la página oficial de la NBA y el precio\_juagdor hemos introducido el precio de los jugadores más importantes y el resto hemos puesto precios aleatorios mediante un script.

-Jugadores Equipo: Entre jugadores y equipos surge una relación mn y por tanto surge una nueva tabla llamada jugadores\_equipos en esta se encuentran los jugadores de cada uno de los equipos.

-Jugadores Libres: Entre ligas y jugadores surge una relación mn y por tanto se crea una tabla llamada jugador\_libre en esta se encuentran todos los jugadores de cada una de las ligas que no tengan un equipo.

-Estadisticas totales: Esta tabla es la suma de los jugadores\_partidos\_reales es decir las estadísticas de un jugador por partido y se suman en esta tabla.

-Partidos reales: En esta tabla se encuentran los partidos que se van a disputar entre los equipos en esta tabla se encuentra la fecha del partido los equipos que los disputan y el resultado.

**Diseño HTML**

Para este proyecto hemos utilizado un html como base y de este se han desarrollado

los demás.

El html base contiene un <nav></nav> superior que contiene el logo de la página

y el menú de esta. Después hay un sectión que contiene el contenido de la página

y por último un footer que siempre es el mismo que dentro tiene una caja con información relevante y enlaces a nuestras redes.

Hemos intentado seguir la regla del triángulo de oro y poner la información mas relevante en la parte superior izquierda de la pantalla, en cuanto los colores hemos utilizado un azul oscuro que no canse a la vista del usuario mientras que para destacar enlaces y botones hemos utilizado un color que contraste bien como el verde fosforito.

Y de esta salen todas las demás, el contenido que cambia en cada página es el section de esta manera es mucho mas facil crear los html y después mediante css hacerlas responsive.

**Diseño CSS**

De la misma manera que en el HTML tenemos un css base del cual hemos ido copiando y pegando en los demás css sobre el <nav></nav>que contiene el logo y el menú y el footer con la información importante

Luego cada una tiene sus propiedades css propias que se adaptan a cada section de cada uno de los html.

**MediaQuerys**

Estas las hemos utilizado basicamente por que hay cambios drásticos entre movil, tablet y ordenador. El gran cambio y mas drástico es el menú de la pagina web el cual en el ordenador se ve fijo en la parte superior de la web, sin embargo, en tablet y movil desaparecen de ahí y se convierten en un menú desplegable pulsando una label.

**Controladores**

Todos nuestros controladores reciben información del front de nuestro proyecto es decir o de las views(html) o de los JS de cada una de estas mediante llamadas Ajax. Estos reciben esa información la procesan y la envian al modelo o core si fuera necesario retornar alguna información al Ajax se llamaría al model se obtendría la información se procesaría y se enviaría como objeto JSON a la llamada AJAX.

**Model o Core**

Estos reciben la información que les da el controlador, la procesan y realizan la consulta necesaria a la base de datos estos recogen la información como un array y la envian al controlador. Cabe destacar que todos los metodos que hacen sentencias SQL por motivos de seguridad son privados y no pueden ser llamados por el controlador asi que estos retornan la información a unos públicos que estos si pueden ser llamados por un controlador y no hay problemas de seguridad.

* 1. Pruebas a nivel modular y global

Hemos hecho distintas pruebas para probar la calidad del proyecto y gracias a estas hemos podido solucionar distintos problemas que hemos encontrado.

* A nivel de BBDD: realizando consultas para comprobar si estaban correctamente relacionadas las tablas de la BBDD.
* A nivel de html y css: gracias a una extensión de chrome llamada window resizer hemos podido encontrar distintos bugs relacionados con las vistas de estas.
* A nivel de Codigo(PHP): gracias a distintos test que hemos realizado hemos podido solventar errores relacionados con busquedas en algunos inputs de ligas

y problemas con la confirmación de la cuenta del usuario.

* A nivel de Codigo(JQuery): Aqui hemos utilizado la famosa técnica ensayo error ya que al tener que realizar llamadas AJAX relacionadas con eventos etc hemos

tenido que ir avanzando según ibamos escribiendo el código y dandonos cuenta de los bugs por el camino.

* 1. Mantenimiento

Debemos mantener un servidor activo para el correcto funcionamiento del proyecto.

Para poder mantener este proyecto es necesario utlizar los scripts que hemos diseñado para meter automáticamente los datos que le pasas por csv a la BBDD, por tanto, para mantener las estadísticas actualizadas debemos hacer un control semanal de estos csv visitando la web de la nba y cogiendo las estadísticas de esta.

Para no sobresaturar el servidor, ni la BBDD debemos cada cierto tiempo vaciar tablas de datos (cada temporada de la nba) y volviendo a llenar estas con los scripts

de importación.

Cada semana debemos utilizar el script que lleva a cabo el cálculo de las puntuaciones de cada equipo y sumar estas puntuaciones a sus respectivos equipos.

Cuando hay algun cambio en cuanto a traspasos entre equipos tendremos que actualizar de forma manual o automática (con algún script de importacion) esos traspasos que realizan los equipos de la NBA.

Cada cierto tiempo cambiar las noticias de la pagina principal y actualizar el twitter de la cuenta oficial de basketmunio.

Crear continuamente nuevas ligas oficiales de basketmunio para que la gente pueda entrar a estas libremente.

Estar pendiente y contestar a los emails que recibamos ya sean por "bugs", mejoras, o simplemente algun tipo de consejo.

Cada cierto tiempo cambiar los escudos de los equipos.

1. Conclusiones
   1. Objetivos alcanzados

Hemos cumplido todos los objetivos expuestos anteriormente excepto:

• Tener un mercado donde se muestren todos los jugadores que no se encuentren en ningún equipo y poder realizar traspasos, tanto fichajes como ventas de tus propios jugadores.

No hemos podido completar este objetivo por falta de tiempo.

* 1. Futuras mejoras

Ofrecer a los usuarios tanto ligas públicas como privadas, estas últimas tendrán que tener una contraseña para poder acceder a ellas.

* Adaptar el proyecto al framework de PHP Laravel.
* Adaptar el proyecto y migrar este a un Hosting.

Tener un mercado donde se muestren todos los jugadores que no se encuentren en ningún equipo y poder realizar traspasos, tanto fichajes como ventas de tus propios jugadores.

1. Referencias bibliográficas

**Confirmacion del login:**

* <http://www.taringa.net/comunidades/webdesign/3519713/Aporte-Groso-Enviar-email-de-confirmacion-al-registrar.html>
* <https://www.fosforito.net/2016/05/21/enviar-emails-php-mail/>
* <http://snipplr.com/view/48702/registrar-usuarios-en-php-con-validacion-de-datos-y-activacin-por-mail--3/>

**Pasar datos de un excel a una BBDD:**

- <https://www.puntogeek.com/2011/06/24/de-excel-a-mysql-facilmente-y-sin-romperte-la-cabeza/>

**Visor CSV:**

- <http://www.becsv.com/csv-viewer.php>

**Informacion para concocer estadisticas:**

- <http://stats.nba.com/>

- <http://www.basketball-reference.com/leagues/NBA_2017_games-october.html>

**Iconos de la web:**

-<http://fontello.com/>

**Fuentes de la web:**

- **Plugin:** //platform.twitter.com/widgets.js

-**Google font:** <https://fonts.googleapis.com/css?family=Roboto>

**Dudas referentes a código:**

-<https://secure.php.net/>

-<https://www.w3schools.com/>

-<https://developer.mozilla.org/en-US/docs/Glossary/jQuery>

-<https://api.jquery.com/>

-[www.forosdelweb.com](http://www.forosdelweb.com/)

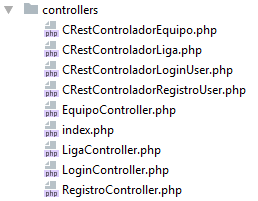
**Git Hub:**

-[github.com](http://github.com/)

1. Anexos

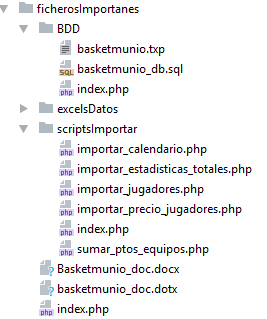
Estructura completa del proyecto (capturas)

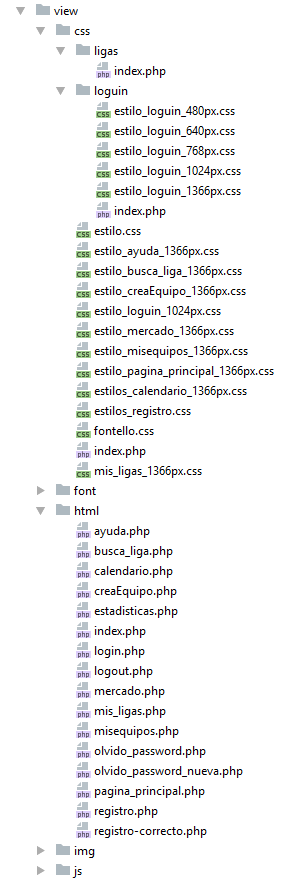
**Controladores**

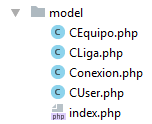


**Ficheros importantes**

\*La carpeta “*excelsDatos*” contiene mas de 30 ficheros csv distintos.



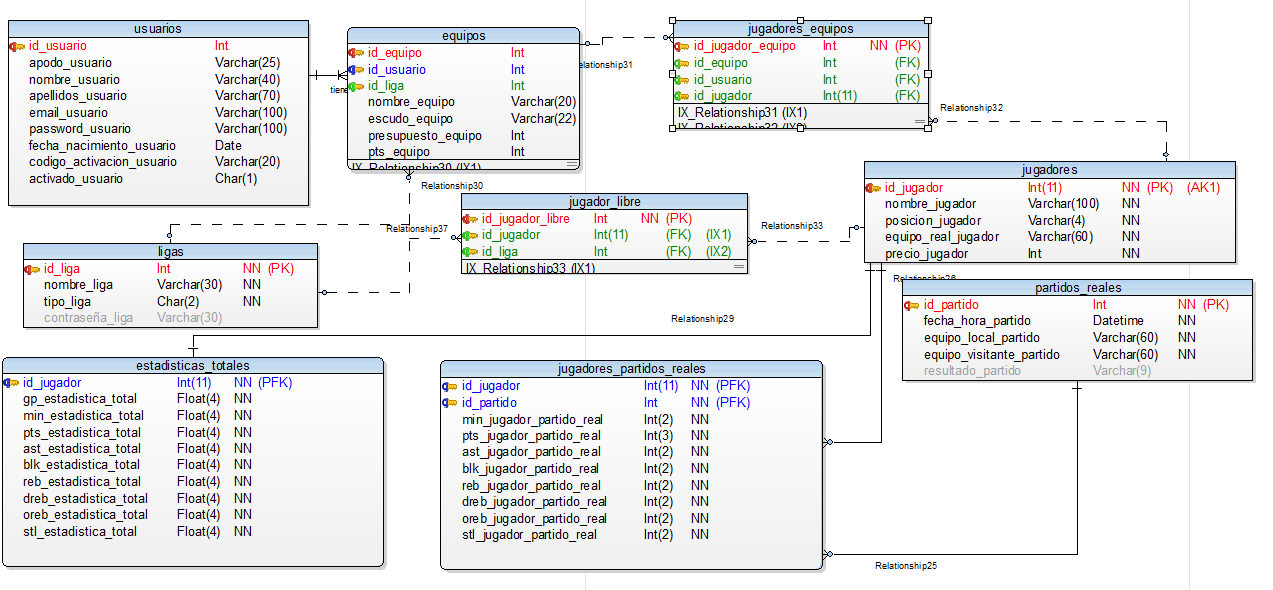
**Modelos** **Vista**



---------------------------------------

Respescto a la vista, carpetas como Font, img o js se encuentran sin desplegar debido a la gran cantidad de ficheros que poseen.

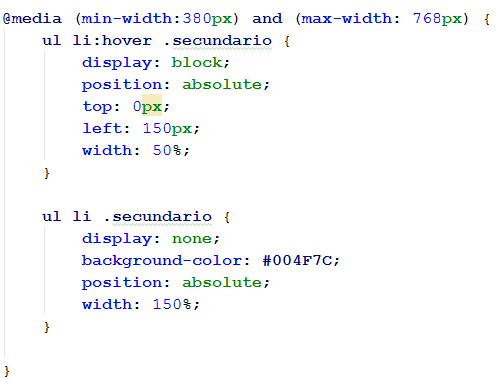
Estructura de la BBDD



Ejemplos

Media query

Ejemplo de media query de la página principal.



HTML

Ejemplo de HTML del menú de la página principal.



Llamada Ajax

Ejemplo de llamada AJAX cuando seleccionamos una opción en un select



Controlador

Ejemplo del controlador y con su constructor



Model

Ejemplo de una función de model

